Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 20 (2013) 454 – 459

Procedia Computer Science

Complex Adaptive Systems, Publication 3
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2013- Baltimore, MD

Finding Semantic Equivalence of Text Using Random Index Vectors

Rosemary D.Paradis[a]*, Jinhong K.Guo[b], Jack Moulton[c], David Cameron[d],Pentti Kanerva[e]

[a]*Lockheed Martin, Information Systems & Global Solutions,VF Bldg 100, PO Box 61511, King of Prussia, PA 19406, USA*
[b]*Lockheed Martin, Advanced Technology Laboratories,3 Executive Campus, Cherry Hill, NJ 08002, USA*
[c]*Lockheed Martin, Information Systems & Global Solutions,VF Bldg 100, PO Box 61511, King of Prussia, PA 19406, USA*
[d]*Lockheed Martin, Information Systems & Global Solutions, PO Box 49041 MS t70 San Jose, CA 95161, USA*
[e]Stanford Center for the Study of Language and Information. Menlo Park, CA 94025, USA

Abstract

The challenges of machine semantic understanding have not yet been satisfactorily solved by automated methods. In our approach, the semantics and syntax of words, phrases and documents are represented by deep semantic vectors that capture both the structure and semantic meaning of the language. Our experiment reproduces the experiment done by Patwardhan and Pedersen 2006, but uses random index vectors for the words, glosses and tweets. Our model first determines random index vectors from glosses and definitions for words from WordNet. From these foundational semantic vectors, random index vectors that represent phrases, sentences or tweets are determined. Our set of algorithms relies on high-dimensional distributed representations, and their effectiveness and versatility derive from the unintuitive properties of such representations: from the mathematical properties of high-dimensional spaces. High-dimensional vector representations have been used successfully in modeling human cognition, such as memory and learning. Our semantic vectors are high-dimensional and capture the meaning of a language expression, such as a word, phrase, query, news article, story or a message. A key benefit of our method is that the dimensionality of the vectors remains constant as we add data; this also allows good generalization to rarely seen words, which "borrow strength" from their more frequent neighbors.

Keywords: high-dimension; random index; machine learning; social networking; computational linguistics; artificial intelligence

1. Introduction

Automated word meaning, semantic understanding and semantic similarity have not yet been satisfactorily solved by automated methods. Determining the meaning of words and the content of documents and messages are difficult challenges for automation, especially when trying to generalize over multiple styles of speaking and over multiple languages. Our methods of creating vectors that contain this information improves the speed and accuracy of algorithms in capturing meaning from free text, thus also improving the tools for sifting through and screening text for further examination by human analysts. Our algorithms are based on high-dimensional distributed

* Corresponding author. Tel.: 1-607-972-1486; fax: 1-607-754-8806;
E-mail address: rosemary.paradis@lmco.com

doi:10.1016/j.procs.2013.09.302

representations, and their effectiveness and versatility derive from the unintuitive properties of such representations---from the mathematical properties of high-dimensional spaces.

High-dimensional vector representations have been used successfully in modeling human cognition, such as memory and learning (i.e. [1], [2], [3], [4] and [5]). They prevail in machine learning and are used widely in corpus-based language research and in practical systems such as search engines. More recently, it has been discovered that syntactic structure and relations can also be encoded in such vectors, and that the vectors for syntax can be combined with vectors that encode semantic similarity. These ideas are based on two ideas that descend from artificial neural networks, *Random Indexing* (RI) and *Holographic Reduced Representation* (HRR) ([6], [7], [8] [9]).

A semantic vector is a high-dimensional vector that captures the meaning of a language expression, such as a word or a phrase or a query or a news article or a story or a message. Expressions with similar meaning are encoded by vectors that are close to each other in the vector space, and can be the means for automating the selection of messages for further examination, for example.

Traditional semantic vectors, such as those produced by Latent Semantic Analysis (LSA) [9], [10], ignore the compositional or deep structure of language---its syntax or grammar---and do not capture much of the meaning in a message. Syntax is traditionally handled with symbolic processing, but then including the semantics becomes a craft. Our method allows syntactic categories and relations to be included in the semantic vectors, and allows the automation of Natural Language Processing (NLP) and the extraction of meaning beyond what is possible with current methods.

1.1. Natural Language Processing

Natural language processing offers two main challenges: Word meanings are highly ambiguous and most words are rare, making them hard to model [11]. Current knowledge-based methods rely on large lexicons, rule-bases, and ontologies. Well-tuned knowledge-based methods have performed very well in recent competitions [12], [13], [14], and [15]. However, coverage of widely varying text is brittle, and knowledge has to be encoded for each language.

Current statistical algorithms (e.g., LSA, probabilistic Latent Semantic Analysis (pLSA) [16], and Latent Dirichlet Allocation (LDA) [17]) model the statistics of word co-occurrence or word sequences (as in Conditional Random Fields (CRF) [18]), and have led to tremendous progress in machine translation and topic discovery, but these methods largely operate on surface features, ignoring syntactic or semantic structure beyond part-of-speech tagging. Scaling using these conventional methods can also be a problem. In contrast, our approach makes use of additional linguistic information, and thus will overcome conventional methods' lack of semantics and scaling.

A unified treatment of language by computer automated algorithms would greatly facilitate interpretation of large collections of text, reports, messages, and news articles that analysts use. Text query and search would be improved, and interacting with computer-controlled systems would be made less difficult. More powerful and inclusive deep semantic vectors would improve many traditional NLP tools.

2. Semantic Vectors

Semantic vectors capture the meanings of words based on the contexts in which they occur. Our method allows a rich variety of contexts, including neighboring words in text, larger contexts such as paragraphs, and "relational contexts" as given by WordNet (http://wordnet.princeton.edu) and other ontologies and semantic nets. The resulting semantic vectors for "similar" words are close to each other, but the nature of the similarity depends on the context used to derive the vectors, and includes both substitutional similarity ("doctor" and "nurse") and relatedness ("doctor" and "hospital"). Our semantic vector methods work equally well using the bag of words in a paragraph or a document, as when using purely local context; and also make it easy to broaden the definition of context to include paragraph-leveltopics and grammatical and causal relations between words.

The use of random indexing as a means of storing and comparing text statements has great potential. There are many ways to compare text. The most common way currently is to use a sliding window around a word and collect the words used with it to help define this word and find the similarity in its use with other words. A second approach is to use all the words in a statement to build a semantic vector which captures the meaning of the word in that context. Capturing this data creates a very large workspace which is difficult to store and process in real time. While many applications do not require real time processing it is very important in many applications. The storage space

required to handle a large text corpus is also a concern with many methods. Random Indexing using sparse vectors is a good approach to overcoming both of these concerns.

A key benefit of our semantic vectors is the substantial dimensionality reduction from projecting very large vocabularies onto, for our example, an eight thousand dimensional space of semantic vectors. This allows good generalization to rarely seen words, which "borrow strength" from their more frequent neighbors, it allows incorporation of larger contexts, and it speeds computation over the more memory intensive n-gram language models. Our algorithm incorporates word vectorization through a novel high dimensional random vector described in Kanerva (2009) [4]. The model we used is similar to the experiment done by Patwardhan and Petersen (2006) [19]. Our semantic vector model representations will scale to very large training corpora and will enable better coverage of rare terms and higher certainty for normal text patterns as well as easy inclusion of additional information to the current baseline. This method is also language agnostic and little effort is required to apply them to new foreign languages.

3. Experiment Description

The experiment we did was based on similar work done by Patwardhan and Petersen [19]. In our experiment, our goal was to build a random index vector that would measure the semantic relatedness of concepts, using WordNet. WordNet represents a large number of concepts and is available in many languages. However, in this experiment we used only English words. Each word in WordNet contains a definition and a set of synonyms that represent the same concept. For example, the concept *home* can also be expressed as *dwelling, domicile, abode, or habitation*, and many others. These words are known as a *synset* in WordNet. Each word is also associated with a definition, as well as a *gloss*. A gloss is a partial sentence that contains the meaning of the work in a certain sense; for example, a gloss of home is “deliver the package to my home”. The definition for this first sense of home is “where you live at a particular time”. Each RI representation of one definition and gloss became a first order vector for each word or concept. The second order vector for a word is created by combining its first order vector with the first order vector of the other concepts.

In order to create our vectors, we start by creating individual vectors for all the words in WordNet. These are all the single words from WordNet – the version we used was the wordnet30 database. For example, for the word *home,* we generate a vector of length 8000 and randomly populate 24 of the positions with a 1 or -1. A RI is generated for each sense of the word. These RI’s are saved in a table. The process starts by looking at all noun words (there could be nouns, verbs, adjectives and adverbs). All noun senses are processed, and for each sense, the glosses and the definition RI’s are added to the original RI by first removing all stop words and parsing the remaining text. Stop words are common, short function words, such as *a*, *at*, *for*, and *the*. We then determined the lemma (the base or dictionary form) for each word, and then found the corresponding RI that was generated from the original WordNet words. Next, the RI’s are added together for each word. This process is then done for verbs, adverbs and adjectives. The RI’s for nouns, verbs, adverbs and adjectives are then added together – this becomes the RI for the total word. This is the second order vector and is created by combining the first order vectors for any particular word, definition and gloss using addition with the first order RI vectors of the words in the definition of the other words in the WordNet definition. The focus words were selected from a list of words that had been used in a number of other experiments to determine context similarity [19], [20] and [21].

3.1. Random Index Example

The following are the steps we took for the creation of the semantic vector for “home” which has definitions for four parts of speech.

1. A RI for the word home is generated. For the definitions and glosses, the stop words are removed and the lemmas are determined for the remaining words; for example, the first modified gloss is *deliver package home*. The next step is to find the RI for each of the words deliver, package, home and add them together. This is the output from WordNet for the first sense of home: home#1 (S: (n) **home#1**, place#7 (where you live at a particular time) *"deliver the package to my home"; "he doesn't have a home to go to"; "your place or mine?")* :.
2. The RI’s for each word as it was originally generated is now added to the RI’s for the synonyms, glosses and definitions. This becomes the RI for home noun sense#1.
3. The same process is done for all the senses for home noun – that would be 9 senses. When complete, all noun RI’s are added together for home noun and saved in the database.
4. This process is then done for verbs, adjectives and adverbs for *home*.

5. The next step is to add the RI for nouns to the RI for verbs, adjectives, and adverbs. This becomes the RI for the word *home*.

For our experiment, our vocabulary was the set of all the different words in WordNet. The initial random index vector was fixed at the start of processing and was basically uncorrelated with other RI vectors. A semantic vector was then generated by combining the information within each word by adding their RI vectors as described above. The surrounding text becomes the context for each word. The cosine similarity is used as a measure of similarity of these context vectors. Our hypothesis was that words with similar meanings would acquire similar semantic vectors.

4.0 Experiment Results

Our results are promising and are shown in table 1. Our results are comparable to the other results from [14] and [15], but there are some reasons why some of our results are different from the other data. We believe that the best result will come from keeping the senses totally separate as well as by having more words, definitions and glosses in the dictionary that is being used as the basis for each word.

Table 1. Word pairs semantic relatedness result compared to rank of [14] and [15]

	R&G Rank	M&C Rank	P,G,M,C Rank
car;automobile	1	2	1
gem;jewel	2	1	1
journey;voyage	3	6	6
boy;lad	4	3	7
coast;shore	5	5	8
asylum;madhouse	6	9	4
magician;wizard	7	7	2
midday;noon	8	1	3
furnace;stove	9	8	12
food;fruit	10	11	17
bird;cock	11	12	20
bird;crane	12	12	21
tool;implement	13	4	11
brother;monk	14	10	5
lad;brother	15	13	9
crane;implement	16	14	18
journey;car	17	15	15
monk;oracle	18	21	26
cemetery;woodla	19	17	13
food;rooster	20	18	23
coast;hill	21	16	10
forest;graveyar	22	19	25
shore;woodland	23	22	14
monk;slave	24	24	16
coast;forest	25	23	19
lad;wizard	26	20	22
chord;smile	27	27	29
glass;magician	28	25	27
rooster;voyage	29	26	24
noon;string	30	26	28

Looking at words like young, youngish, youngest there are very different senses to young and youngest for example. The Stanford package produces a lemma that is most like the WordNet approach. A proper stemmer often reduces words to a stem that is not in WordNet. To reduce the ~200,000 words in WordNet to 20,000 might be the result of a stemmer. Using a stemmer available that more closely matches WordNet results that may be the best approach. What words to remove or keep is also a question that we experimented with.

Adding in hyponyms, hypernyms, etc. may have boosted the word space also. There are several variables to experiment with. Different linkages could be explored as potential data to add to the method. Some other traditional text processing may be of help also. For example it would be useful to know if we were dealing with a compound sentence in a tweet we were analyzing. When processing word pairs as we did in this experiment it was not a consideration but in tweets it would be.

5.0 Summary

Random indexing can capture context-based semantics in high-dimensional vectors and can complement or replace traditional systems that rely on principal components (e.g., LSA). HRR makes it possible to encode compositional structure in high-dimensional vectors and thus include syntax, which is traditionally handled in

symbolic representation. The vector for a word captures its "meaning" by effectively averaging all the information contained in the contexts in which the word occurs. In the random indexing method, each word in the vocabulary is represented by two high dimensional random indexing vectors. These RI vectors are fixed at the start and uncorrelated with other RI vectors. We then encode the RI vectors into (dense) semantic vectors, or semantic context vectors, that are learned from a large amount of text.

In the RI approach, the structure and ordering of a language is captured through permutation and projection of these random indexing vectors, *flattening* a sequence of vectors `a b c d ...' into a single vector. We have started with labeled English text and English WordNet as sources, and we have developed a method of computing semantic vectors that encode both the context and relations expressed as the network structure. We used a WordNet-based measure of semantic relatedness that combines the structure and content of WordNet with co-occurrence information derived from raw text. Using co-occurrence information along with the WordNet definitions, we built gloss vectors that correspond to each concept in WordNet. The gloss vector measure works by forming second-order co-occurrence vectors from the glosses or WordNet definitions of concepts. Numeric scores of relatedness have been assigned to a pair of concepts by measuring the cosine of the angle between their respective gloss vectors. We showed that this measure compares favorably to other measures with respect to human judgments of semantic relatedness, and that it performs well when used in a word-sense disambiguation algorithm that relies on semantic relatedness. This measure is flexible in that it can make comparisons between any two concepts without regard to their part of speech. In addition, it can be adapted to different domains, since any plain text corpus can be used to derive the co-occurrence information.

References

1. Bower, G. (1967). A multicomponent theory of the memory trace. *Psychology of Learning and Motivation:* 1:229-325.
2. Burgess, C. (2001). Representing and resolving semantic ambiguity: A contribution from high-dimensional memory modeling. In: Gorfein, D.S. (Ed.), On the Consequences of Meaning Selection: Perspectives on Resolving Lexical Ambiguity. APA Press, 2001.
3. Jones, M., Kintsch, W. and Mewhort, D.J. (2006). High-dimensional semantic space accounts of priming, *Journal of Memory and Language* 55.534-552.
4. Kanerva, P. (2009). Hyperdimensional Computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation* 1(2):139-159. http://redwood.berkeley.edu/pkanerva/papers/kanerva09-hyperdimensional.pdf
5. De Vine, L. and Bruza, P. (2010). Semantic oscillations: encoding context and structure in complex valued holographic vectors. In *Proc. AAAI Fall Sym. on Quantum Informatics for Cognitive, Social, and Semantic Processes (QI 2010)*, 11:13 November 2010, Arlington, Virginia.
6. Plate, T. (1991). Holographic Reduced Representations: Convolution Algebra for Compositional Distributed Representations, In the *Proceedings of the 12th International Joint Conference on Artificial Intelligences*, Edited by J. Mylopoulos and R. Reiter, Morgan Kaufmann, San Mateo, CA.
7. Plate, T. (1995). Holographic Reduced Representations. *Neural networks, IEEE transactions on*, *6*(3), 623-641.
8. Plate, T. (2003). *Holographic Reduced Representation: Distributed representation for cognitive structures*. CSLI Publications, Stanford, CA.
9. Kanerva, P., Kristoferson, J. and Holst, A. (2000). Random Indexing of text samples for latent semantic analysis. In Gleitman, L.R. and Josh, A.K. (eds.) Proc. *22nd annual conference of the Cognitive Science Society.* Mahwah, NJ: Erlbaum; p. 1036.
10. Landauer, T. and Dumais, S. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review 104(2):211-240*.
11. Zipf, G.K. (1949). *Human Behavior and the Principle of Least Effort*. Cambridge, Massachusetts: Addison-Wesley.
12. Dolan, W. and Brockett, C. (2005). Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing* (IWP2005), pp. 9–16, Jeju island, Korea.
13. Bentivogli, L, Clark, P., Dagan, I., Dang, H., and Giampiccolo, D. (2011). The Seventh PASCAL Recognizing Textual Entailment Challenge. In *Notebook papers and results*, *Text Analysis Conference (TAC)*. http://www.nist.gov/tac/publications/2011/papers.html

14. Pradhan, S., Ramshaw, L., Marcus, M., Palmer. M., Weischedel, R. and Xue, N. (2011). CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes. *In Proceedings of the Fifteenth Conference on Computational Natural Language Learning (pp. 1-27*), Association for Computational Linguistics, Portland, Oregon.
15. Jurgens, D., Mohammad, S., Turney, P. and Holyoak, K. (2012). Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*, Association for Computational Linguistics, Montreal, Canada.
16. Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, *42*(1-2), 177-196.
17. Blei, D.M., Ng, A.Y., & Jordan, M.I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993-1022.
18. Lafferty, J., McCallum, A., & Pereira, F.C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning (ICML) 2001*.
19. Patwardhan, S., & Pedersen, T. (2006). Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together* (Vol. 1501, pp. 1-8).
20. Miller, G.A. and Charles, W.G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes,* 6(1):1-28, 1991.
21. Rubenstein, H. and Goodenough, J.B. (1965). Contextual correlates of synonymy. *Communications of the ACM,* 8(10):627-633, October 1965.